DEEP LEARNING BASED TEXT REGRESSION

by

Neşat Dereli

B.S., Electronics and Communication Engineering, İstanbul Technical University,
2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering
Boğaziçi University

2019

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor, Prof. Murat Saraçlar for his support and mentorship. I am especially thankful to him for letting me have the freedom and responsibility for the thesis work while supporting me with his invaluable guidance. I would also like to thank the jury members, Assoc. Prof. Arzucan Özgür and Assist. Prof. Ebru Arısoy Saraçlar for kindly accepting to review the thesis and participate in the defense of the thesis.

I thank also the members of Boğaziçi University Signal and Image Processing Laboratory (BUSIM). A special gratitude goes to Alican Gök, Bolaji Yusuf and Can Altay for sharing their experiences. I also appreciate all feedback and suggestions from my friends, Çağrı Aslanbaş, Ilgaz Çakın, Ömer Adıgüzel and Zeynep Yiyener.

The numerical calculations reported in this thesis were performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources). Without TRUBA resources, the experiments described in this thesis work would not have been possible to finish. I wish TUBITAK ULAKBIM continues to support academic research and studies.

I am deeply grateful to Erol Bilecik, the former president of Alumni Association of the İstanbul Technical University and to Sabancı Foundation for their scholarship support during my bachelor's study. Thanks to their financial support, I was able to focus on my education which led me to my master's study. I would also thank Arçelik and Lifemote for supporting me during my master's study and letting me attend my courses.

Finally, I would like to express my warm gratitude to my parents and to my sister for supporting me throughout the years of my study and motivating me continuously. This accomplishment would not have been possible without their encouragements.

ABSTRACT

DEEP LEARNING BASED TEXT REGRESSION

Most financial analysis methods and portfolio management techniques are based on risk classification and risk prediction. Stock return volatility is a solid indicator of the financial risk of a company. Therefore, forecasting stock return volatility successfully creates an invaluable advantage in financial analysis and portfolio management. While most of the studies are focusing on historical data and financial statements when predicting financial volatility of a company, some studies introduce new fields of information by analyzing soft information which is embedded in textual sources. Forecasting financial volatility of a publicly-traded company from its annual reports has been previously defined as a text regression problem. Recent studies use a manually labeled lexicon to filter the annual reports by keeping sentiment words only. In order to remove the lexicon dependency without decreasing the performance, we replace bag-of-words model word features by word embedding vectors. Using word vectors increases the number of parameters. Considering the increase in number of parameters and excessive lengths of annual reports, a convolutional neural network model is proposed and transfer learning is applied. Experimental results show that the convolutional neural network model provides more accurate volatility predictions than lexicon based models.

ÖZET

DERİN ÖĞRENME TABANLI METİNSEL REGRESYON

Finansal analiz metotları ve portföy yönetim tekniklerinin çoğu, risk sınıflandırma ve tahminlemeye dayanır. Hisse senedi getirisinin dalgalanma derecesi bir şirketin finansal riskiyle ilgili güçlü bir göstergedir. Bu sebeple, hisse senedi getirisinin dalgalanma derecesini başarılı bir şekilde öngörmek finansal analiz ve portföy yönetiminde çok değerli bir avantaj yaratır. Bu konudaki araştırmaların çoğu, bir şirketin finansal dalgalanma derecesini tahmin etmek için geçmiş veriler ve şirket bilançosuna odaklanırken bazı araştırmalar ise metinsel kaynakların içerisindeki teknik olmayan bilgileri analiz ederek yeni bilgi kaynakları sunuyor. Halka açık bir şirketin yıllık raporlarındaki metinlerden, o şirketin finansal dalgalanma derecesini öngörmek önceden metin regresyon problemi olarak tanımlanmıştı. Son yapılan araştırmalarda, yıllık raporlardan duygu ifade etmeyen kelimeleri eksiltebilmek için el ile etiketlenmiş bir deyimcelik kullanılıyor. Performansı düşürmeden deyimcelik ihtiyacını ortadan kaldırmak için metin öznitelikleri yerine kelime özniteliklerini kullandık. Yani metinleri, içerlerinde geçen kelimelerle ifade etmek yerine metinlerdeki kelimeleri öznitelik vektörleriyle ifade ettik ve bu durum parametre sayısını arttırdı. Parametre sayısındaki artış ve yıllık raporların aşırı uzunlukları göz önünde bulundurularak evrişimli sinir ağları modeli önerildi ve transfer öğrenmesi uygulandı. Deneysel sonuçlar, evrişimli sinir ağları modelinden alınan dalgalanma derecesi tahminlerinin, deyimcelik tabanlı modellerden alınan tahminlere göre daha yüksek doğrulukta olduğunu gösteriyor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

f	Non-linear function
g	Convolution feature
K	Dimension of word embedding
L	Vocabulary size
M	Length of document
n	Convolution kernel size
N	Number of documents
$\mathbb{N}$	Set of all natural numbers
R	Number of features of convolution layer
$\mathbb{R}$	Set of all real numbers
w_i	ith word in a document

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
ANN	Artificial Neural Network
CBOW	Continuous Bag-of-Words
CNN	Convolutional Neural Networks
CSRR	Corporate Social Responsibility Reports
CRSP	Center for Research in Security Prices
CV	Computer Vision
DMAA	Dimethylamylamine
DL	Deep Learning
FDA	Food and Drug Administration
IR	Information Retrieval
LOG1P	Log Normalized Word Frequency
MD&A	Management's Discussion and Analysis
ML	Machine Learning
MSE	Mean Square Error
MT	Machine Translation
NASDAQ	National Association of Securities Dealers Automated Quotations
NER	Named Entity Recognition
NLP	Natural Language Processing
NYSE	New York Stock Exchange
POS	Part of Speech
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recursive Neural Networks
SVR	Support Vector Regression
TC	Term Count
TF	Term Frequency

TFIDF	Term Frequency Inverse Document Frequency
WRDS	Wharton Research Data Services

1. INTRODUCTION

Languages are used for communication by telling a story, asking a question or giving an order. They are not only the main component of human to human communication but also the main component of human to computer communication. Considering their naturalness properties, languages can be grouped into two types, natural languages also known as ordinary languages and artificial languages.

Natural languages are the languages, evolved naturally through human evolution and history, such as English, German and Turkish. Artificial languages are languages which are fully or partly constructed by human [1]. Computer programming languages, language of mathematics which includes mathematical equations, and language of chemistry which includes chemical formulas are some examples to artificial languages. Most of the Natural Language Processing (NLP) applications provide a mapping, e.g., from a natural language to another one, from a long text to a sentiment class, from a word sequence to an entity class.

Since naturally evolved languages are often vague and the same sentence can be interpreted differently by different people, the evaluation of NLP tasks is nontrivial. Thus, NLP tasks are mostly focused on a partial problem instead of covering the domain as a whole. There are various NLP applications. Machine Translation (MT) maps a sequence of words from one language to another [2]. Named Entity Recognition (NER) maps word chunks to entities, for instance, people, locations, and organizations [3]. Part of Speech (POS) taggers use contextual information to assign word classes such as noun, adjective and verb [4]. Sentiment analysis is used to measure the sentiment polarity of the source which can be a document, a sentence or a phrase [5]. Document summarization is compressing a document into a shorter text or just a sentence by keeping its main message [6]. Question answering is retrieving the answers of questions from a large document [7].

Text regression task is defined by Kogan *et al.* [8] as predicting real-world continuous values by using associated text documents. They used the textual information of annual financial reports to predict the share price volatility of publicly-traded companies. Because text regression task does not require any manual labeling, the evaluation of text regression task is more precise and easier compared to document summarization, sentiment analysis and question answering.

Most of the NLP tasks suffer from the ambiguity of labeling. The same sentence can be interpreted differently by distinct readers. However, text regression is an important test-bed for NLP research [8]. There are also other works which showed the impact of the text regression such as, analysing websites for disease tracking [9], movie revenue forecast [10], author age prediction [11], predicting election results from social media [12].

Kogan defined the problem, forecasting financial volatility from annual reports, as a text regression task and other studies contributed to the task because of its value [13–15]. There are also alternative soft information sources which are used for financial forecast like news [16–19], online forums [20, 21], blogs [22] and bank reports [23]. However, annual reports are more informative and contain less noise since they are regulated by the government. On the other hand, annual reports are not suitable for short-term forecasting.

1.1. Motivation

Government-mandated financial reports contain a large amount of information about firms and their value. However, they are long and dense which makes it costly to analyze each one of the government-mandated financial reports. Forecasting the financial risk of a publicly-traded company by using the annual report of the target company is clearly a valuable information for any investor and any financial portfolio manager. In the present financial world with a great amount of mandated disclosures, the importance of extracting useful insights to make correct decisions increases highly.

The stock return volatility of a publicly-traded company is a fundamental indicator of the stability of the company. It is used as a financial risk indicator and it is essential for investment decisions and financial portfolio management. Forecasting stock return volatility which is used as a financial volatility measure has gained an important attention during the last three decades [15]. Nonetheless, studies which aim to predict the stock return volatility of a publicly-traded company focus mostly on the hard information of the company which is the sum of all quantitative information.

Financial hard information of a company includes financial statements and historical market prices of the company. On the other hand, soft information of a company contains news about the company and textual financial reports of the company. Using soft information to predict financial volatility of a company is introduced as a text regression task [8]. Studies which focus on soft information to present a solution for the text regression task show that including soft information can improve the models which depend on hard information only. Since the task is a new task and initiated in last decade, it attracts computational language researchers and finance researchers.

Financial text regression task can also be compared with other NLP tasks. Most of the NLP tasks require human expertise to label the sentences, the documents or the phrases. However, the financial text regression task is not subject to any human knowledge and manual work because all the labels are available as historical market data.

Furthermore, financial reports and historical data are publicly available. Therefore, growing amount of freely available financial data increases the importance to design an algorithm which can extract the valuable information from the available data. These properties of the task motivates us to research a model which contributes to the financial text regression task.

1.2. Contributions of the Thesis

In this thesis, the lexicon dependency of financial text regression models are removed. The models which are proposed in the previous studies focus on the sentiment polarity of words and phrases while mapping the report to a stock return volatility value. Therefore, they highly depend on a financial sentiment lexicon which is used to extract words with greater sentiment polarity. Nonetheless, human expertise, intuition and manual work are required to create the lexicon. In this work, the proposed models do not require any lexicon and thus the lexicon dependency is removed.

In this work, financial sentiment lexicon is replaced with word embeddings. Some of the previous studies also use word embeddings. However, in the previous studies word embeddings are used only to expand the financial sentiment lexicon. Word embeddings are not included to the model. Our model which is used to predict the financial risk of a company using its annual reports benefits from word embeddings.

Removing the sentiment lexicon and including the word embeddings increase the number of parameters. Artificial Neural Network (ANN) is used to handle the increase of the parameters. The models which are presented in the previous studies are Machine Learning (ML) models. To the best of our knowledge, this is the first work which uses ANN model for the financial text regression task.

There are also previous works which states that ANN models are experimented but the ANN models are not published because they perform worse than the ML models which are presented in that work. Another contribution of this work is achieving better performance with ANN models compared to a ML model which is presented as the best model in a previous study.

1.3. Organization of the Thesis

The structure of the following chapters of this thesis is as follows. Chapter 2 provides background knowledge about related works and technical details. The chapter

includes both previous works which contribute to financial text regression task and previous works which use Convolutional Neural Networks (CNN) for NLP. In Chapter 3, available datasets for financial text regression task and their structure are reviewed. The data structure includes annual reports of publicly-traded companies and financial measures. Financial measures of a company can be calculated using historical market data of the company.

Chapter 4 of the thesis contains system architecture and model architecture. System architecture describes the algorithm by consecutive system boxes. The model architecture presents layered architecture of the ANN model. In Chapter 5, experimental setup is presented, differences of tested models are described, results are reviewed and analysis are discussed.

2. BACKGROUND

This chapter provides background work on financial text regression and CNN. It is important to understand the current state of the domain. Without the background of the previous research, contribution of this work may be unclear. This chapter is divided into two sections. First section provides information about the problem, defined previously. In the same section, literature review, mathematical explanations and further details are provided. In the second section, related works on the model we proposed for the financial text regression problem are presented. Finally, further background about CNN which is used to build the model proposed in this thesis is provided.

2.1. Financial Text Regression

In this section, financial text regression publications are reviewed. Later, the methods, which are used to solve the financial text regression task, are described.

2.1.1. Related Work

Kogan *et al.* [8] published their research in 2009 which forecasts company risk by using annual report of the company. To the extent of our knowledge, their work is the first work that uses NLP for the defined problem. Stock return volatility, which is explained in Section 2.1.2, is used as the risk indicator of a company. They collected the annual reports of publicly-traded US companies and use only the Item 7A Management's Discussion and Analysis (MD&A) of the annual reports.

All the numerical values are replaced with the same placeholder and the rest of the words are stemmed. Then, each report is presented with its bag-of-words features where three different bag-of-words features are used. These features, which are explained in Section 2.1.3, are term frequency (TF), term frequency inverse document frequency (TFIDF) and log normalized word frequency (LOG1P). Later, support vec-

tor regression (SVR) is used to train a regression model which predicts the stock return volatility using the bag-of-words features of the annual reports.

In the same study, SVR is used for supervised learning tasks and its impact is greater when dimension of feature space is larger than number of examples [24–26]. Mean square error (MSE) is used to evaluate the result. MSE can be calculated using Equation 2.1 where y_i is the stock return volatility value of a company and $\hat{y}_i$ is the stock return volatility prediction for the same company.

$$\textbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.1}$$

Furthermore, the stock return volatility of the previous year is used as an additional feature. The results of different models are presented for the years between 2001 and 2006.

Wang *et al.* [13] made use of a sentiment lexicon, which is prepared for financial documents [27]. The sentiment lexicon is explained in Section 2.1.5. They used the same text resource, used by Kogan *et al.* [8], which is the MD&A section of the US companies' annual report for the same years. However, only sentiment words, which are specified in financial sentiment lexicon, are used and rest of the words in the annual reports are filtered out.

In the same work, they also presented a ranking task. Risk ranking is useful for portfolio management. LOG1P is used for regression task and TFIDF is used for ranking task. The work showed improvement and thus it is stated that the sentiment lexicon is useful while using bag-of-words features.

Tsai and Wang [14] expanded the financial sentiment lexicon via Continuous Bag-of-Words (CBOW) which is a Word2Vec model, described in Section 2.1.4 [28]. They also built a syntactically richer CBOW model which contains POS tag of each word. Cosine similarity is used to extract top-n nearest word for each keyword in the financial

lexicon. Extracted words are used to construct the expanded financial lexicon. Word vectors are not used to train SVR but only to expand the lexicon. LOG1P is used for regression task and TFIDF is used for ranking task. In contrast to previous works, stock return volatility value of previous year is not used as a feature while training the SVR model.

In 2017, Rekabsaz *et al.* [15] presented a research where feature fusion is used and companies of distinct sector are analyzed separately. Instead of using MD&A section of annual reports, Item 1A - Risk Factors is used. Contrary to previous works, reports of recent years, 2006 to 2015, are used. It is shown that reports of consecutive three to four years are more similar to each other than reports of temporally separate years. Thus, reports of the most similar years, 2012 to 2015, are used for training the model and to evaluate the model. Training and test data are split using 5-fold methodology instead of a temporal folding. Furthermore, BM25 is used as an aditional bag-of-word feature which is explained in Section 2.1.3.

In the same study, in addition to early fusion implementation in previous works, Rekabsaz *et al.* [15] implemented Multi Kernel Learning (MKL) and stacking. MKL is known as intermediate fusion and stacking is known as late fusion [29, 30]. SVR with linear kernel which is used in previous research is changed with SVR with Radial Basis Function (RBF) since it performed better. They used r^2 metric (square of the Pearson correlation coefficient) and MSE as evaluation metrics. r^2 metric is also known as coefficient of determination. r^2 metric is shown below in Equation 2.2 where $\bar{y}$ is the mean of stock return volatility values and $\bar{\hat{y}}$ is the mean of stock return volatility predictions.

$$\mathbf{r^2} = \left(\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \right)^2 \tag{2.2}$$

They also experimented an ANN model to test the effectiveness of automatic feature learning. Nonetheless, they could not achieve an ANN model which is comparable to the SVR model.

2.1.2. Stock Return Volatility

Stock return volatility is defined as standard deviation of adjusted daily closing prices of a target stock over a period of time [8, 31]. The target period of time can be chosen between two events where each event contain information related to prediction. While annual reports are used as information source, stock return volatility is calculated between annual report release dates. Adjusted daily closing price is calculated by including any corporate action such as stock splits, stock offerings and dividend declarations to determine a value that occurred before the market opening of next day. Let S_t be the adjusted closing stock price for the day t. Then, stock return for the day t is given as follows (Equation 2.3):

$$R_t = \frac{S_t}{S_{t-1}} - 1 \tag{2.3}$$

Stock return volatility $v_{[t-\tau,t]}$ for τ days prior to t is given as follows where $\bar{R}$ is the mean of the stock return values (Equation 2.4):

$$\mathbf{v_{[t-\tau,t]}} = \sqrt{\sum_{i=0}^{\tau} \frac{(R_{t-i} - \bar{R})^2}{\tau}} \tag{2.4}$$

Stock return volatility is a solid indicator of the financial risk of a company. Therefore, forecasting stock return volatility successfully creates an invaluable advantage in financial analysis and portfolio management.

2.1.3. Bag-of-words Features

Bag-of-words features are used in NLP and information retrieval (IR) to describe a sentence or document [32]. A matrix is constructed where an axis contains documents and other axis contains words. Words of a source can be represented in different ways such as binary term count, term count (TC), term frequency (TF). Binary term count of following two sentences are given in Table 2.1.

- *Sentence 1:* A dog is running around a tree.
- *Sentence 2:* The dog crashed into the tree.

Table 2.1. Binary term count.

	a	**around**	**crashed**	**dog**	**into**	**is**	**running**	**the**	**tree**
Sentence 1	1	1	0	1	0	1	1	0	1
Sentence 2	0	0	1	1	1	0	0	1	1

Term count, $\mathbf{TC}(w_i; d_j)$, denotes the number of occurrence of ith word in document j. Term count of above presented two sentences is shown in Table 2.2

Table 2.2. Term count.

	a	**around**	**crashed**	**dog**	**into**	**is**	**running**	**the**	**tree**
Sentence 1	2	1	0	1	0	1	1	0	1
Sentence 2	0	0	1	1	1	0	0	2	1

Other bag-of-words feature representations, TF, TFIDF, LOG1P, BM25, are given in Equation 2.5, Equation 2.6, Equation 2.7 and Equation 2.8:

$$\mathbf{TF}(w_i; d_j) = \frac{\mathbf{TC}(w_i; d_j)}{|d_j|} \tag{2.5}$$

TFIDF is used as a statistical measure which indicates importance of the word in a document.

$$\mathbf{TFIDF}(w_i; d_j) = \frac{\mathbf{TC}(w_i; d_j) \log\left(\frac{N}{|\{d:TC(w_i;d_j)>1\}|}\right)}{|d_j|}, \quad \forall j \in \{1, ..., N\} \tag{2.6}$$

LOG1P is log normalized version of **TF**.

$$\mathbf{LOG1P}(w_i; d_j) = \log\left(1 + \mathbf{TC}(w_i; d_j)\right) \tag{2.7}$$

BM25 is also known as Okapi BM25 where BM stands for best matching. **BM25** is used by search engines to measure the relevance of a word to a document.

$$\mathbf{BM25}(w_i; d_j) = \frac{(k+1)\overline{tf(w_i; d_j)}}{k + \overline{tf(w_i; d_j)}}, \quad k \in \mathbb{R} \tag{2.8}$$

where k is a parameter and Equation 2.9 is used to calculate Equation 2.8:

$$\overline{\mathbf{tf}(w_i; d_j)} = \frac{\mathbf{TC}(w_i; d_j)}{(1-b) + b\frac{|d_j|}{\mathbf{avgdl}}}, \quad b \in \mathbb{R} \tag{2.9}$$

b is a parameter and average document length **avgdl** is defined in Equation 2.10:

$$\mathbf{avgdl} = \frac{1}{N}\sum_{j=1}^{N} |d_j| \tag{2.10}$$

Bag-of-words features can also be built using consecutive words instead of a single word. They are called bag of bi-grams, tri-grams, four-grams etc. Bag-of-words features are effective for document classification. However, grammar and word sequence of the source is ignored and bag-of-words feature matrix is sparse. Therefore, word embedding models are better at including information of word order.

2.1.4. Word Embedding

Word embedding is a method which is used to represent words with vectors to embed syntactic and semantic information [28]. One-hot encoding representation also maps words to vectors but one-hot vectors are sparse. The dimension of one-hot vector is equal to the size of vocabulary, all words used for the model. On the other hand, word embedding vectors are dense and thus they are continuous. Mostly, word embedding vector dimension is a few hundred whereas vocabulary size is tens of thousands.

Each value of a word embedding vector contains information about a feature of the corresponding word [33]. Different dimension indexes represent different features.

The information kept by a dimension index depends on the model which is used to build word embedding vectors. In [34], it is stated that word embedding represented words in similar context are mapped to near points in vector space. The vectors of the words *dog* and *cat* will be near to each other. Furthermore, word relations are reflected in word embeddings where $queen - king \approx woman - man$. The major impact of using word embedding on NLP models is inserting word relation information into the model.

Word2Vec is a method to construct word embeddings [35]. It is a neural network based model and contains two different models which are CBOW model and continuous skip-gram model [36]. CBOW model tries to find an optimum vector representation which can predict a word by using preceding and following words in a sentence or document.

On the contrary, skip-gram model tries to predict preceding and following words for a given word. Since word embedding models are trained on large datasets, efficiency is very important. Word2Vec models use negative-sampling to improve the efficiency. There are also other word embedding models such as GloVe and fastText [37, 38].

2.1.5. Financial Sentiment Lexicon

Lexicon is a finite set of lexemes of a particular language, domain, person etc. [39]. A lexeme is a word or phrase with a precise sense [40]. Lexicons differ from dictionaries since dictionaries focus on grammatical forms and include multiple form of a same lexeme. Loughran and McDonald [27] presented financial sentiment lexicon. It contains six different word list which reflects sentiment polarity in financial context. It is stated that, Harvard Pyschosociological Dictionary presented in [41] which is commonly used for sentiment classification performs worse in finance documents.

Around 75% of negative word list (Harvard-IV-4 TagNeg (H4N)) contained in Harvard Pyschosociological Dictionary does not have any negative polarity in financial context. Table 2.3 shows difference of Harvard Pyschosociological Dictionary and Financial Sentiment Lexicon by 10 examples of 3 categories. Each word, presented in

the table, is contained in one of the two lists only. Details of each set of Financial Sentiment Lexicon are as follows:

(i) *Fin-Neg:* words having negative polarity in finance (e.g. downsize, penalty, resign).

(ii) *Fin-Pos:* words having positive polarity in finance (e.g. achieve, boost, gain).

(iii) *Fin-Unc:* words denoting uncertainty but with emphasis of weak precision instead of focusing on risk (e.g. almost, could, vary).

(iv) *Fin-Lit:* litigious words (e.g. claimant, interlocutory, tort).

(v) *MW-Strong:* strong modal words (e.g. always, must, will).

(vi) *MW-Weak:* weak modal words (e.g. may, nearly, possible).

Table 2.3. Difference between Harvard Pyschosociological Dictionary and Financial Sentiment Lexicon.

Harvard Pyschosociological Dictionary			**Financial Sentiment Lexicon**		
Negative	**Positive**	**Weak**	**Negative**	**Positive**	**Weak**
anger	assist	absent	acquit	boom	almost
dirt	care	alone	antitrust	despite	appearing
empty	charm	blind	bankrupt	empower	could
fat	comfort	blur	boycott	innovate	may
gun	famous	cease	censure	loyal	might
hate	glorious	decay	delist	lucrative	nearly
hunt	humor	drop	enjoin	outperform	perhabs
ruin	joy	flee	illegal	revolution	possible
sad	joke	tiny	nullify	smooth	sometimes
ugly	warm	weak	postpone	strength	suggest

2.2. Convolutional Neural Networks

In this section, CNN for NLP tasks are reviewed and layers of CNN models are described. CNN models are originally introduced for computer vision (CV) problems. Recently, they are shown to be effective also for NLP tasks such as question answering [42], semantic query retrieval [43] and sentence modelling [44].

2.2.1. Related Work

Kim [45] showed that using CNN models for NLP classification tasks can outperform ML and Recursive Neural Networks (RNN) models. The CNN model consists of an embedding layer, a convolution layer, a max-over-time pooling layer and a fully connected layer respectively. Dropout and softmax is employed after the fully connected layer.

In the same study, model variations are experimented by changing the embedding layer. Embedding layer can be chosen as static or non-static, initialized randomly or pretrained, constructed as single channel or multiple channels. The CNN model is evaluated using different datasets of various tasks such as sentiment analysis and question classification.

Bitvai and Cohn [46] presented a CNN model to solve a text regression problem where the model tries to predict continuous data instead of finite labels. The model is used to predict movie revenues by using review texts and movie metadata. The CNN architecture differs from [45] by adding multiple fully connected layers and removing both dropout and softmax layers. Moreover, in [46], there is a single dataset which includes movie reviews, metadata and weekly revenue.

2.2.2. Embedding Layer

In NLP, it is common to present each word with an incremental id number. Mostly, id numbers are assigned after vocabulary words are sorted by number of oc-

currence in a decreasing order. Embedding layer of ANN models maps each word identity to a vector. Vector dimension is fixed and specifies number of word features. If embedding layer is not static, vectors are updated during model training.

Updating word vectors leads to similar words to be grouped to near points in vector space and to separate words with distinct properties. Similarity for each model and problem may differ. For example, a model trained with a POS dataset groups syntactically similar words whereas a model trained with a sentiment classification dataset groups semantically similar words. Embedding layer can be initialized using pretrained word embeddings to include extra word features or to speed up word feature acquisition.

2.2.3. Convolution Layer

In computer vision, dimension of convolution layer is equal to dimension of the image. If the image is a two dimensional (2D) image, it is common to use a 2D convolution layer. However, in NLP, one dimensional (1D) convolution is used since sentences, paragraphs, documents etc. are 1D. Let each word of a document be represented using word vectors of K dimension as $w_i \in \mathbb{R}^K$. The document of size M can be represented by concatenating all words:

$$w_{1:M} = w_1 \oplus w_2 \oplus ... \oplus w_M, \quad w_{1:M} \in \mathbb{R}^{KM} \tag{2.11}$$

where $\oplus$ is concatenation operator. A convolution layer applies convolution operation on each word window which has size of n. Each convolution feature, $g_{1:M-n+1} \in \mathbb{R}^{M-n+1}$, is calculated by using a kernel, $weight \in \mathbb{R}^{KN}$, and a bias, $bias \in \mathbb{R}$ [47]:

$$g_{1:M-n+1} = g_1 \oplus g_2 \oplus ... \oplus g_{M-n+1} \tag{2.12}$$

$$g_j = weight \cdot w_{1:M} + bias \tag{2.13}$$

Figure 2.1 shows the 1D convolution of a document with a single feature output which is defined in Equation 2.13

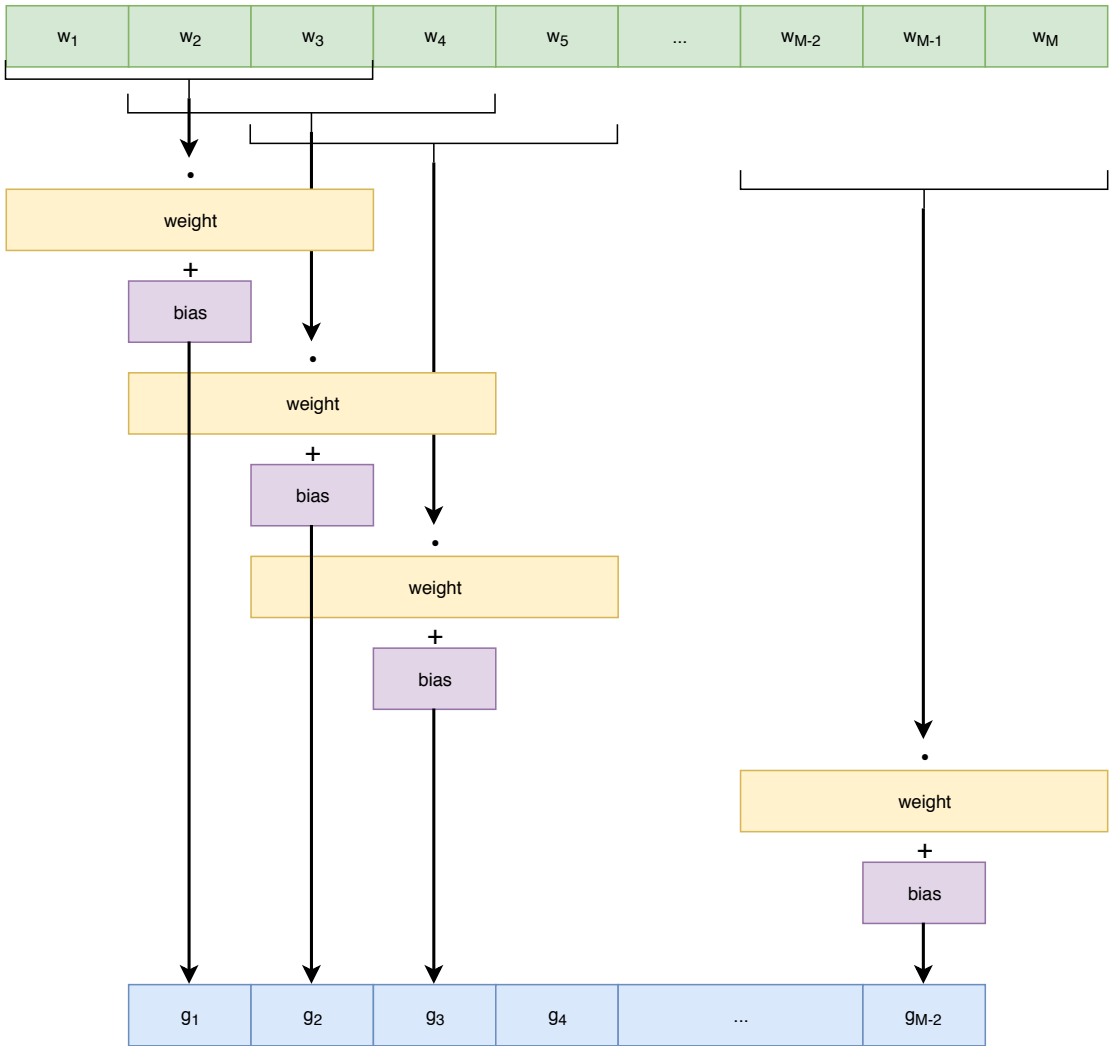

Figure 2.1. 1D Convolution.

It is also common to use a non-linear function f at the output of convolution layer where convolution feature becomes:

$$g_j = f(weight \cdot w_{1:M} + bias) \tag{2.14}$$

f can be sigmoid, tanh and ReLU (Rectified Linear Unit) etc.

2.2.4. Max-over-time Pooling Layer

Max pooling layer is used to focus on dominant features in CV. Since images are 2D and 2D convolution is used, max pooling is applied to local blocks which are smaller than the image size. However, sentences are 1D and their length are not fixed. Thus max-over-time pooling is used which returns maximum value over a sequence [48]:

$$max(g_{1:M-n+1}) = max(g_1, g_2, ..., g_{M-n+1}) \tag{2.15}$$

If there are multiple features, max-over-time pooling returns a distinct maximum value for each feature. Figure 2.2 shows the max-over-time pooling layer, applied to multiple feature sequences, which is defined in Equation 2.15

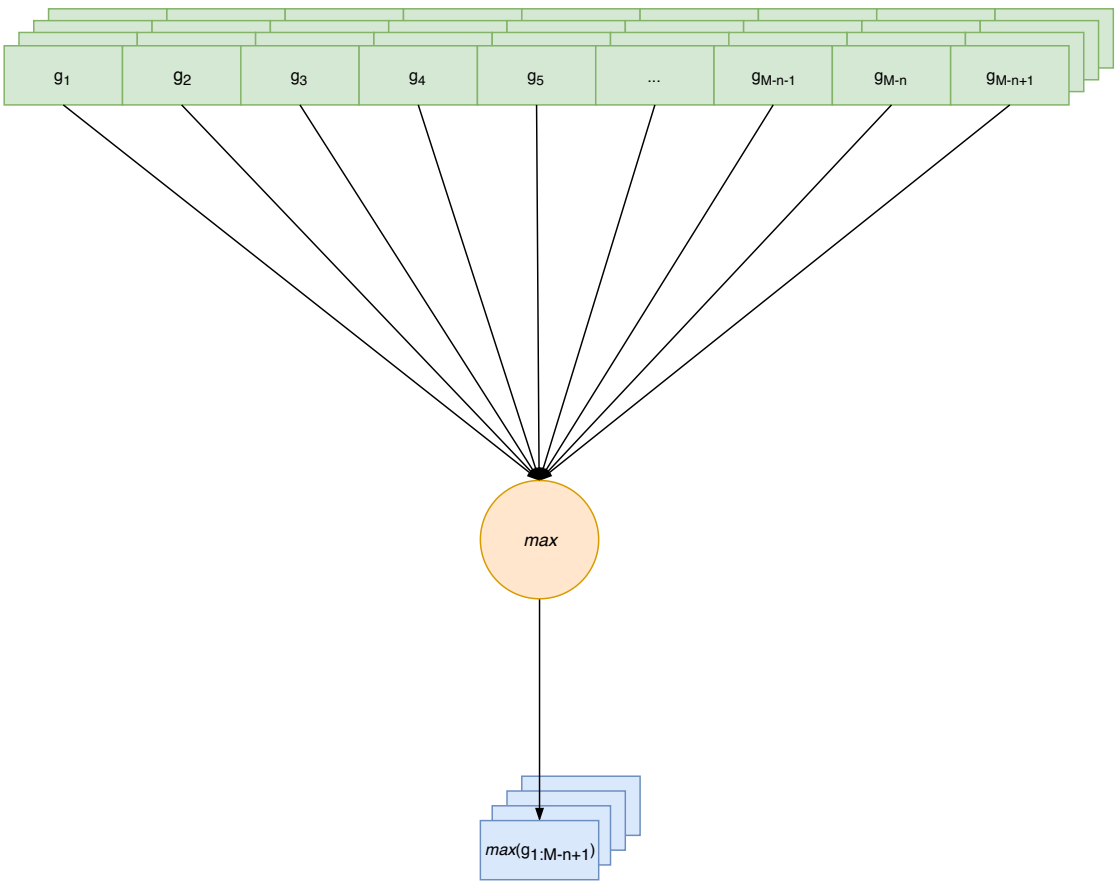

Figure 2.2. Max-over-time Pooling.

3. DATA

This chapter provides the datasets which are related to the work and further explanation about sources and targets. Deep learning (DL) and ML based systems can be divided into two broad groups which are supervised and unsupervised models. Unsupervised learning models try to investigate the data by extracting its feature, finding its distribution and clustering the data points [49, p. 103].

On the other hand, supervised learning models are more straightforward. They are trained using data points and labels to predict targets. Labels and targets can be finite or continues. Finite labels are tags which describe the data points. Continuous labels, on the other hand, are a portion of map where all labels sum up to a complete system.

Supervised learning methods are heavily rely on annotated datasets. Dataset annotation can be manual which is expensive or automatic. Since the model quality and performance are affected by the size of the dataset, problems with automatic annotation are more eligible for ML and ANN methods. Conversely, manual labor annotation required problems rely on the quality of the annotation work and thus noise is introduced in the dataset.

Financial text regression problem is a task where manual labeling work is not required. Data points of the problem can be phrases, sentences, documents etc. The problem is defined as building a model which can map these data points to a numerical value presented in the market. By using the company identity a text source and financial measure can be linked without any extra labeling work.

Financial text can be anything which includes language content about any financial topic. Some financial text sources are social media posts, news, annual reports and social responsibility reports [50]. This work focuses on annual reports, especially annual reports which are in the US 10-K report format which is described in Section 3.1.

Each report is mapped to a continuous financial measure which is described in Section 3.2 in details. In this work stock return value is used as data point label. Stock return value represents risk information of the company.

3.1. 10-K Reports

In U.S., annual report filings, known as 10-K reports, are mandated by the government in a strictly specified format. 10-K reports are available on the U.S. Security Exchange Commission (SEC) Electronic Data Gathering, Analysis and Retrieval (EDGAR) website [51]. In [52], it is stated that 10-K fillings are the most precise and complete single document of financial information available to investors.

10-K reports often consist of crucial information about performance of the company and its financial state. However, there is always delay between the date when filings are prepared and the date when fillings are published. The time delay depends on company size, audit report complexity and internal control [53].

The EDGAR system was introduced in 1984 by SEC to motivate certain firms to experiment with electronic reports. Until 1993, electronic filing was voluntary but after 1993 they become mandatory for all firms. In 1997, almost all hard paper fillings were eliminated. The SEC also published amendments, in 2002, to shorten the filling deadlines from 90 to 60 days. However, general format has not been changed since 1995 when the full phase-in was completed.

10-K reports consist of 4 parts which include 15 items and conclusion section. Most informative items of 10-K reports are Item 1A - “Risk Factors”, Item 7 - “Management’s Discussion and Analysis of Financial Condition and Results of Operations” and Item 7A - “Quantitative and Qualitative Disclosures about Market Risk” [54].

3.1.1. Item 1A - Risk Factors

Item 1A informs investors about the most important risks that concern the company or its securities. Risks are generally listed in order of importance. This item does not provide the company perspective but only focuses to present the risks. Presented risks can be the risks which affect the entire economy, company's industry, geographic region. Also risks which are unique to the company can be presented.

First two paragraphs of Item 1A - Risk Factors section from 10-K report of AAR Corp. released in 17 July 2006 is shown in Figure 3.1.

> The following is a description of some of the principal risks inherent in our business. The risks and uncertainties described below are not the only ones facing us. Additional risks and uncertainties not presently known to us, or that we currently deem immaterial, could negatively impact our results of operations or financial condition in the future.
>
> We may be affected by continuing problems in the aviation industry. As a provider of products and services to the aviation industry, we are greatly affected by the overall economic condition of that industry. The aviation industry is historically cyclical. Early in calendar year 2001, the commercial aviation industry began to experience the negative effects of a worldwide economic downturn. The events of September 11, 2001 exacerbated that condition, resulting in a significant decline in air travel and reduced capacity by most of the major U.S.-based airlines. Since September 11, 2001, the aviation industry has been also negatively affected by historically high fuel prices, the war on terrorism and the outbreak of Severe Acute Respiratory Syndrome, or SARS. As a result of these and other events, certain customers filed for bankruptcy protection, including Air Canada, Aloha Airlines, Delta Air Lines, Mesaba Airlines, Northwest Airlines, U.S. Airways, United Airlines and Varig.

Figure 3.1. First 2 paragraphs of AAR Corp. 2006 10-K report 1A Item.

3.1.2. Item 7 - Management's Discussion and Analysis

Item 7 contains company's perspective on its performance of the past fiscal year. In MD&A section, company management tells its story in its own words. Correctness of this section and the projectile for the next year affect decision of investors. In MD&A section operations of the company, financial view summary, liquidity of the company, trends, uncertainties and business risks are provided. Some examples are given below:

- Consumer companies - Discussion about meeting the change of customer tastes.
- Manufacturing companies which use natural resources - Discussion about handling commodity risks and scheduling resource management.
- Global companies - Discussion about managing exchange rate risks.
- Financial institutions - Discussion about handling liquidity and adequate capital assurance under different circumstances.
- Technology firms - Discussion about laws and regulations compliance or their impact.

First paragraph of MD&A section from 10-K report of AAR Corp. released in 26 August 2002 is shown in Figure 3.2.

3.1.3. Item 7A - Quantitative and Qualitative Disclosures about Market Risk

Item 7A presents information about how the company can be affected by market risks, such as exchange rate risk, equity price risk, interest rate risk, commodity price risk. Plans about managing company's market risk exposures may be discussed.

Quantitative and Qualitative Disclosures about Market Risk section from 10-K report of AAR Corp. released in 26 August 2002 is presented in Figure 3.3.

The Company's future operating results and financial position may be adversely affected or fluctuate substantially on a quarterly basis as a result of the difficult commercial aviation environment exacerbated by the September 11, 2001 terrorist attacks and the events that followed, the relatively weak worldwide economic climate and other factors, including: (1) decline in demand for the Company's products and services and the ability of the Company's customers to meet their financial obligations to the Company, particularly in light of the poor financial condition of many of the world's commercial airlines; (2) lack of assurance that sales to the U.S. Government, its agencies and its contractors (which were approximately 25.5% of total sales in fiscal 2002), will continue at levels previously experienced, since such sales are subject to competitive bidding and government funding; (3) access to the debt and equity capital markets to finance growth, which may be limited in light of industry conditions and Company performance; (4) changes in or noncompliance with laws and regulations that may affect certain of the Company's aviation related activities that are subject to licensing, certification and other regulatory requirements imposed by the FAA and other regulatory agencies, both domestic and foreign; (5) competitors, including original equipment manufacturers, in the highly competitive aviation aftermarket industry that have greater financial resources than the Company; (6) exposure to product liability and property claims that may be in excess of the Company's substantial liability insurance coverage; (7) difficulties in being able to successfully integrate future business acquisitions; (8) fluctuating market values for aviation products and equipment in the current aviation environment; (9) difficulty in re-leasing or selling aircraft and engines that are currently being leased on a long or short-term basis and (10) environmental proceedings as described in Item 3.

Figure 3.2. First paragraph of AAR Corp. 2002 10-K report MD&A section.

3.2. Financial measures

Financial measures are continuous numerical values which can be calculated using the stream data of a market. For the text regression task each financial measure is calculated for one year since 10-K reports are annual. Four different financial measure

The Company's exposure to market risk includes fluctuating interest rates under its unsecured bank credit agreements, foreign exchange rates and accounts receivable. See Item 7 "Critical Accounting Policies" for a discussion on accounts receivable exposure. During fiscal 2002 and 2001, the Company did not utilize derivative financial instruments to offset these risks.

At May 31, 2002, $70,485 was available under credit lines with domestic banks under revolving credit and term loan agreements, and $1,795 was available under credit agreements with foreign banks (credit facilities). Interest on amounts borrowed under the credit facilities is LIBOR based. As of May 31, 2002, the outstanding balance under these agreements was $40,500. A hypothetical 10 percent increase to the average interest rate under the credit facilities applied to the average outstanding balance during fiscal 2002 would have reduced the Company's pre-tax income by approximately $108 during fiscal 2002.

Revenues and expenses of the Company's foreign operations in The Netherlands are translated at average exchange rates during the year and balance sheet accounts are translated at year-end exchange rates. Balance sheet translation adjustments are excluded from the results of operations and are recorded in stockholders' equity as a component of accumulated other comprehensive income (loss). A hypothetical 10 percent devaluation of foreign currencies against the U.S. dollar would not have a material impact on the financial position or results of operations of the Company.

Figure 3.3. Item 7A of AAR Corp. 10-K report released in 2002.

can be used where each one contains different information about the company [55]. These financial measures are:

- *Post-Event Return Volatility:* Root mean square error (RMSE) of an adjusted stock price from Fama-French three-factor model [56]. Fama-French three-factor model predicts expected return successfully using size of the company and book value of the company. Difference between Fama-French three-factor model prediction and real value is a valid risk measure.

- *Stock Return Volatility:* In Section 2.1.2, stock return volatility is defined. It can be calculated using Equation 2.4. Stock return volatility is a risk measure which presents vulnerability of the stock price against external impacts.
- *Abnormal Trading Volume:* Average trading volume of 4 day event window in which volume is normalised using average and standard deviation of last 2 months. There are three different approaches for abnormal trading volume [57]. Datasets, described in Section 3.3, use the definition above from [27].
- *Excess Return:* Difference between buy and hold return of a target stock and buy and hold return of a target market index.

Financial measures can be calculated using historical databases of stock market such as Center for Research in Security Prices (CRSP) and Yahoo Finance [58]. CRSP is used widely since it also provides automatic measure calculations. Furthermore, researchers and professional investors rely on CRSP for its accuracy. It is accessible via third-party partners such as Wharton Research Data Services (WRDS). However, WRDS and other third-party partners provide paid services only. On the other hand, Yahoo Finance is freely available.

3.3. Dataset Variants

Preparing a dataset for financial text regression problem requires an annual report source and a historical database of stock market. For U.S. companies annual reports can be collected from SEC EDGAR and historical stock prices or historical finance measures can be collected from CRSP or Yahoo Finance. SEC EDGAR provides annual report search from ticker symbol of a U.S. company.

A ticker symbol, also known as stock symbol, is an abbreviation which is used to identify a particular stock on a stock market. A stock symbol can consist of combination of letters, numbers or both. For example, the stock symbol F identifies publicly traded shares of Ford Motor Company in New York Stock Exchange (NYSE). Ticker symbols on NYSE have up to three letters.

On the other hand, National Association of Securities Dealers Automated Quotations (NASDAQ) listed stocks have four letter stock symbols. AAPL, AMZN, MSFT, NFLX which are some stock symbols on NASDAQ correspond to Apple, Amazon, Microsoft and Netflix respectively. Searching AAPL in SEC EDGAR returns all 10-K reports of Apple Inc. Similarly, searching AAPL in CRSP returns all historical stock price records, available.

Although accessing an annual report using ticker symbol is easy, accessing a ticker using an annual report is not always possible using SEC EDGAR. Therefore, ticker symbols of publicly traded companies can be used as master identifiers.

A financial text regression dataset of U.S. companies can be built by choosing publicly traded companies and a time window. Reports and stock prices can be collected by using ticker symbols of chosen companies. Time window values which are start date and end date can be chosen as the release date of the report and one year after the release date of the report respectively. Financial measures can be calculated from collected stock prices. Four different datasets are presented below.

3.3.1. 10-K Corpus

10-K Corpus [59] is presented in [8]. The dataset includes 10-K reports and stock return volatility as financial measure of U.S. companies. SEC EDGAR is used to collect 10-K reports and CRSP is used to calculate stock return volatility values. The dataset also contains MD&A section separately. MD&A section is extracted from the complete report. 10-K reports are collected after 1995 since full phase-in for 10-K reports are completed in 1995.

Number of reports are shown in Table 3.1 for each year. Reports are recorded by release date and thus a company can have two annual reports recorded in the same year. For example, Parametric Technology Corp published two 10-K reports in 2003. One of them is published in 28 January 2003 and the other one is published in 24 December 2003. However, it has not published any report in 2002. The reason of releasing two

reports in a single year is violation of deadline. Therefore, Parametric Technology Corp released notification of inability to timely file form 10-K in 31 December 2002.

Table 3.1. Reports by year in 10-K Corpus.

Year	**Number of Reports**
1996	1408
1997	2260
1998	2462
1999	2524
2000	2425
2001	2596
2002	2846
2003	3612
2004	3559
2005	3474
2006	3308

3.3.2. Extended 10-K Corpus

Extended 10-K Corpus [60] is described in [55]. Similar to 10-K Corpus, Extended 10-K Corpus also uses SEC EDGAR to collect 10-K reports and CRSP via WRDS to calculate financial measures. Four different financial measures are used: post-event return volatility, stock return volatility, abnormal trading volume and excess return which are described in Section 3.2.

Extended 10-K Corpus also provides reports of later years which are 2007 to 2013. Number of reports are shown in Table 3.2 for each year. However, reports of 1996 to 2006 are not same as 10-K Corpus. It can be seen by comparing report counts in Table 3.1 and Table 3.2. Extended 10-K Corpus removed duplicate reports and some short reports. Therefore, number of reports differ between 1996 and 2006 compared to 10-K Corpus.

Table 3.2. Reports by year in Extended 10-K Corpus.

Year	**Number of Reports**
1996	1203
1997	1705
1998	1940
1999	1971
2000	1884
2001	1825
2002	2023
2003	2866
2004	2861
2005	2698
2006	2564
2007	2495
2008	2509
2009	2567
2010	2439
2011	2416
2012	2406
2013	2336

3.3.3. Financial Volatility Dataset

In [15], instead of focusing on MD&A section, it is focused on "Risk Factor" section. Therefore, Financial Volatility Dataset [61] contains 10-K reports between 2006 and 2015. After 2005, SEC mandated companies to include a "Risk Factor" section in their annual reports to review the most significant factors increases their

risk [54]. Thus, the dataset does not include reports before 2006. Different risk factors can be presented in Item 1A such as regulation changes, potential lawsuits, competition risks and financial condition risks [62].

SEC EDGAR is used to collect 10-K reports but in contrast to previous works, Yahoo Finance is used to calculate financial measures. Stock return volatility is used as financial measure. Furthermore, sector of each company is provided in the dataset. Table 3.3 lists number of reports for each year.

Table 3.3. Reports by year in Financial Volatility Dataset.

Year	**Number of Reports**
2006	646
2007	664
2008	697
2009	800
2010	863
2011	927
2012	887
2013	959
2014	1051
2015	1090

3.3.4. JOCo Corpus

JOCo Corpus [63] is presented in [50]. It has two major difference from earlier released datasets. JOCo Corpus does not contain U.S. companies only but also German and U.K. companies. Moreover, it includes corporate social responsibility reports (CSRR). However, the corpus does not contain any financial measure. Annual reports (AR) are collected from company website investor relations section. Therefore, reports does not contain any release date but a correspondence year. In U.S., it corresponds to the year after 10-K report is released.

JOCo contains 30 most intensively traded and most highly valued, 30 middle-sized and 30 technology companies from 3 different countries. Stock market indexes are used to select companies from 3 different categories. DAX, MDAX and TexDAX are used for Germany. FTSE, FTSE 250 and FTSE techMARK are used for U.K. Dow Jones, S&P 500 and NASDAQ 100 are used for U.S.

More details are provided in Table 3.4. JOCo is a free dataset but in contrast to other datasets, JOCo is not publicly available. Download link of a copy is provided after a form is filled to confirm that the corpus will be used for academic purpose.

Table 3.4. Reports by year in JOCo Corpus.

	Germany		**UK**		**US**	
Year	**AR**	**CSRR**	**AR**	**CSRR**	**AR**	**CSRR**
2000-	31	10	23	6	13	1
2000	49	5	40	5	50	1
2001	55	10	46	11	55	4
2002	61	9	55	14	60	8
2003	61	15	61	17	59	10
2004	64	13	66	19	62	13
2005	69	19	74	20	66	16
2006	71	16	81	23	69	22
2007	76	23	88	25	72	28
2008	77	22	87	26	76	28
2009	80	25	87	27	77	29
2010	83	25	88	31	76	39
2011	83	32	89	30	78	41
2012	85	35	88	36	81	42
2013	87	39	87	36	82	44
2014	87	35	88	37	86	49
2015	90	39	0	0	24	12

4. METHODOLOGY

In this chapter, the implementation of our base model and improvements are described. After literature review and data exploration, it is important to choose a dataset, a performance measure and reference model to compare performance. They are described in the first section. In the second section, the system architecture is described which provides a high level presentation of the system from dataset to financial measure prediction. Finally, in the last section, the deep learning model architecture is presented.

4.1. General Information

In Section 3.3, available datasets for financial text regression task are described. Since data amount is important while deep learning models are used, Extended 10-K Corpus is chosen. It is described in Section 3.3.2 and contains more than 1000 reports for each year from 1996 to 2013. Python [64] is used as programming language and PyTorch is used as deep learning framework. Performance of the trained models is measured using MSE, coefficient of determination and Spearman's rank correlation coefficient which are defined in Section 4.2.4.

4.2. System Architecture

The system architecture of the application, designed to solve the financial text regression task consists of four main blocks. These are data loader, preprocessing, model training and model evaluation which are presented in Figure 4.1. It can be seen as a system that takes textual report inputs and provides financial measure forecasts. The system does not contain evaluation of a trained model only but it also includes model training phase. The details of the model is described in Section 4.3.

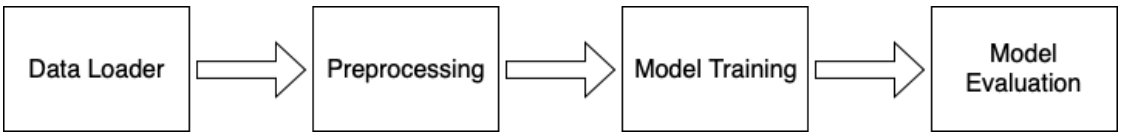

Figure 4.1. System Architecture.

4.2.1. Data Loader

Extended 10-K Corpus contains complete reports, MD&A sections, tokenized MD&A sections and financial measures in seperate folders. MD&A section and tokenized MD&A section folders include reports in discrete folders where each folder represents a year. On the other hand, financial measure folders such as stock return volatility folder includes data of all years in discrete files.

Each file in the financial measure folders contains stock return volatility values of all reports, belong to a target year. For example, "2002.logvol.+12.txt" file includes the natural logarithm values of stock return volatility for each report which is released in 2002. The stock return volatility value of a company for a year represents risk of the company for a time window of one year which starts from release date of the report. Other financial measure values are stock return volatility for the previous 12 months, abnormal trading volume, excess return and post-event return volatility.

Our data loader creates structured tables for each year using pandas library of python. Each element of the table has report ID, natural logarithm of stock return volatility for following 12 months, abnormal trading volume, excess return and natural logarithm of post-event return volatility values which correspond to logvol pos, abnormal, excess and logfama shown in Table 4.1 respectively.

Report id column is also used to locate the report in the file system. For example, the complete report file name and the mda section file name of the report 000360206-10-K-20020319 are 000360206-10-K-20020319.full and 000360206-10-K-20020319.mda respectively.

Table 4.1. Report ID to financial measures subset.

report id	logvol pos	abnormal	excess	logfama
000360206-10-K-20020319	-3.51312	-0.426236	0.081201	-3.346716
000361105-10-K-20020826	-2.96153	0.446931	0.011236	-3.144642
00086T103-10-K-20020221	-3.0751	0.196639	0.144895	-2.940609
00088E104-10-K-20020412	-2.8439	1.174327	-0.039362	-2.893518
00088U108-10-K-20020401	-2.34899	-0.360171	0.004018	-1.331952
00089C107-10-K-20020729	-2.77063	-0.078733	0.062897	-2.898859
001031103-10-K-20020129	-3.15641	-0.184233	-0.020328	-3.16976
001296102-10-K-20020416	-2.01508	0.382288	0.133271	-0.888752
001303106-10-K-20021018	-1.87637	0.172007	0.462904	-0.185592
001957505-10-K-20020401	-3.41161	-0.425611	-0.029099	-2.672649

Structure table is used to link textual reports which are located in the file system separately with financial measures. In our work, natural logarithm of stock return volatility is used as the financial measure. Next step is creating batch where each sample in the batch contains a textual report and a stock return volatility as the label of the report.

The MD&A sections of textual reports are used as the text source and they are preprocessed which is described in Section 4.2.2. Then, tokens of preprocessed MD&A sections are mapped to token identifiers. Finally, after all these processes the batch is used to train the model and evaluate it which are described in Section 4.2.3 and in Section 4.2.4 respectively.

4.2.2. Preprocessing

The MD&A section of the 10-K reports in the Extended 10-K Corpus are already tokenized by removing punctuation, replacing numerical values with # and downcasing the words. In the previous studies, reports are stemmed by using the Porter stemmer

[65]. In this work, MD&A section of reports are also stemmed by Natural Language Toolkit (NLTK).

Stemming is used to map words to their roots. Mapping words to their roots is beneficial to group words and decrease word variations. However, stem of a word does not have to be meaningful. There is also another method which is used to convert words to their roots. It is called lemmatization. Lemma of a word which corresponds to meaningful root of a word is always an actual word.

In this work, stemming is used instead of lemmatization because word embeddings trained by Tsai *et al.* [55] maps stems to word vectors. Another reason of choosing stemming over lemmatization is comparability. Previous works which are presented in Section 2.1.1 also use stems instead of lemmas to convert words to their roots. Another advantage of using stemming and lemmatization is decreasing the vocabulary size of the word embeddings and thus reducing the parameters of the model.

4.2.3. Model Training

Details of the DL model is described in Section 4.3. The DL model includes an embedding layer, explained in Section 2.2.2. Instead of random initialization of embedding layer of the model, initialization with pretrained word embeddings enables the model to capture contextual information faster and better. In our work, we used pretrained word embeddings provided in [55]. They used MD&A section of 10-K reports from 1996 to 2013 to train the word embeddings with a vector dimension of 200 by using word2vec continuous bag-of-words (CBOW).

The model is trained during multiple epochs. At each epoch the complete batch is supplied as input to the model where the batch contains reports and stock return volatility of multiple years. Each epoch consist of multiple iterations and at each iteration the model gets a minibatch whic is a subset of batch, as an input.

Each input report in the batch contains word indices which are mapped to word vectors in embedding layer. After the model training is complete, weights and bias values of all layers are updated. Using the model with updated weights and bias values, stock return volatility of a company can be predicted by using an annual report of the target company.

MSE is used as performance measure during the training which means that the model optimizes MSE. There are also other performance measures which can be used to observe the improvement of the model. Coefficient of determination and Spearman's rank correlation coefficient are some of them and they can be used to measure the performance of a text regression model.

In our work only MSE is optimized during training and other two performance measures are only used to evaluate the trained models. Coefficient of determination and Spearman's rank correlation coefficient is described in Section 4.2.4 in details.

4.2.4. Model Evaluation

MSE is chosen as the main evaluation metric which is formulated in Equation 2.1. Our model optimizes MSE during the training. On the other hand, coefficient of determination and Spearman's rank correlation coefficient are used to evaluate trained models.

Both coefficient of determination and Spearman's rank correlation coefficient are based on Pearson's correlation coefficient. Coefficient of determination is the square of Pearson's correlation coefficient. On the other hand, Spearman's rank correlation coefficient is Pearson's correlation coefficient of rankings.

Pearson's correlation coefficient is used to measure correlation of two set of variables. It is bounded between -1 and 1. Having a Pearson's correlation coefficient near to 0 means two set of variables are independent. A positive Pearson's correlation coefficient points to linear correlation. When Pearson's correlation coefficient is near 1,

distribution of the two set of variables are near.

If Pearson's correlation coefficient is negative on the other hand, then the two set of variables have negative correlation. It means that the values of one set increases while the values of the other one decreases and vice versa. When Pearson's correlation coefficient is near to -1 then amount of decrease and increase are very close to each other. Pearson's correlation coefficient can be calculated by:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{4.1}$$

where X and Y represent two set of variables.

Coefficient of determination is the square of the Pearson's correlation coefficient. It does not provide any information about direction of the correlation. However, it provides the strength of dependency of the two set of variables. The coefficient of determination is also used as a evaluation metric in [15].

Detailed equation of coefficient of determination is provided in Equation 4.2:

$$\mathbf{r^2} = \left(\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \right)^2 \tag{4.2}$$

Target stock return volatility set y and predicted stock return volatility set $\hat{y}$ is used to calculate the coefficient of determination. Its simple form is as follows:

$$r^2 = \left(\frac{\text{cov}(\hat{y}, y)}{\sigma_{\hat{y}} \sigma_y} \right)^2 \tag{4.3}$$

Spearman's rank correlation coefficient is a measure which is used to evaluate the ranking performance of a model. Real volatility values and predicted volatility values can be used to calculate Spearman's rank correlation coefficient. Each set contains samples which consist of a company identifier and the volatility value of the company.

The Spearman's rank correlation coefficient of two sets is equal to the Pearson's correlation coefficient of the rankings of the sets. The rankings of a set can be generated by sorting the volatility values of the set in an ascending order and enumerating them. The rankings of a set contains samples which consist of a company identifier and a volatility rank of the company. Spearman's rank correlation coefficient of the sets y and $\hat{y}$ which are target stock return volatility and predicted stock return volatility respectively can be calculated by:

$$\rho_{\hat{y},y} = \frac{\mathrm{cov}(rank_{\hat{y}}, rank_y)}{\sigma_{rank_{\hat{y}}}\sigma_{rank_y}} \tag{4.4}$$

where $rank_{\hat{y}}$ and $rank_y$ represent the rankings of the sets $\hat{y}$ and y respectively.

4.3. Deep Learning Model Architecture

The architecture of our base network is presented in Figure 4.2 which is similar to previous works that use CNN for NLP [45–47]. Before reports are fed into the embedding layer, their lengths are fixed to M words and reports with less than M words are padded. The output matrix of the embedding layer, $E \in \mathbb{R}^{KM}$, consists of K-dimensional word vectors where the unknown word vector is initialized randomly and the padding vector is initialized as zero vector. Each element of the word vector represents a feature of the word.

The convolution layer consist of different kernel sizes where each kernel size represents a different n-gram. Figure 4.2 shows tri-gram, four-gram and five-gram examples. Let $n \in \mathbb{N}$ be the kernel width of a target n-gram. Each convolution feature $f_i^c \in \mathbb{R}^{M-n+1}$ is generated from a distinct kernel weight, $weight_i^n \in \mathbb{R}^{Kn}$, and bias, $bias_i \in \mathbb{R}$. Rectified linear unit (ReLU) is used as the non-linear activation function at the output of the convolution layer,

$$f_{ij}^c = ReLU(weight_i^n \cdot w_{j:j+n-1} + bias_i) \tag{4.5}$$

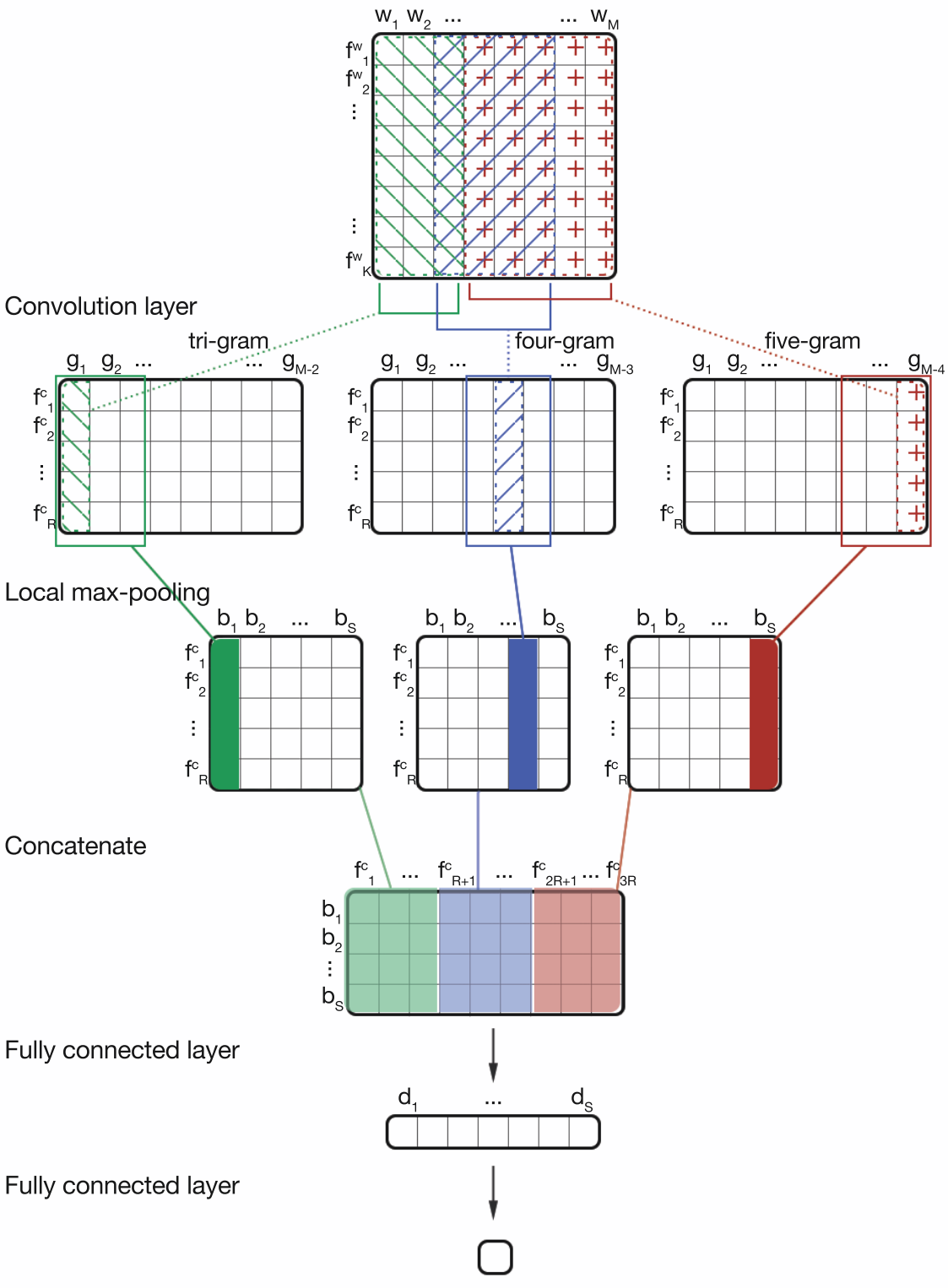

Figure 4.2. Deep Learning Model Architecture.

Note that the convolution features, f_i^c have $M - n + 1$ dimensions and they contain different information than word features, f_i^w. Convolution features are concatenated as

$$f_i^c = [f_{i1}^c, f_{i2}^c, ..., f_{i+n-1}^c] \tag{4.6}$$

g_i in Figure 4.2 represents each n-gram element thus there are $M-n+1$ n-gram elements for each individual n-gram. Next step is local max-pooling layer which basically applies max-over-time pooling to smaller word sequence instead of the complete text [48]. Each sequence length is h and there are S outputs for each sequence,

$$b_i = \max(g_{ih:i(h+1)-1}) \tag{4.7}$$

where $b_i \in \mathbb{R}^R$. After the local max-pooling layer is applied to all convolution layer output matrices, they are merged by concatenating feature vectors. Later, dropout is applied to the merged matrix and finally it is fed into two sequential fully connected layers.

5. EXPERIMENTS AND RESULTS

In this chapter, our experiments are explained and the results are presented. First, the setup for the experiments are described. Then, various models which perform better than the other models and tested for all years, are explained. Later, the evaluation results are presented for various performance measures. Finally, evaluation results and trained models are analyzed.

5.1. Setup

The hyper-parameters of the CNN models are decided by testing them with our baseline CNN model. All weights of the baseline model are non-static and randomly initialized. Final hyper-parameters are shown in the Table 5.1. The details of fixed text length M, embedding vector size K, convolution layer output features size R and convolution layer kernel sizes n are described in Section 4.3.

Table 5.1. Hyper-parameters of the model.

M (Fixed Text Length)	20000
K (Embedding Vector Size)	200
R (Convolution Layer Output Features Size)	100
n (Convolution Layer Kernel Sizes)	{3, 4, 5}
Mini-batch Size	10
Local Max-pooling Window Size	200
Dropout Probability	0.5
Learning Rate	0.001

Kogan *et al.* [8] showed that using reports of the last two years for training performs better than using reports of the last 5 years. Rekabsaz *et al.* [15] presented

similarity heat-map of ten consecutive years and stated that groups consist of three to four consecutive years are highly similar. Our experiments also show that including reports which are four years older than test year into the training set does not always help and sometimes even causes noise.

In this work, the reports of three consecutive years were used for training while reports of the last year were used for validation to determine the best epoch. After the best epoch is determined, it is used as fixed epoch and the oldest year is ignored while the first step is repeated to train a new network without using validation set but fixed epoch instead. For example, reports of 2006 to 2008 are used as training set while reports of 2009 is used for validation. If the best result is achieved after 30 epochs, a new network is trained with reports of 2007 to 2009 through 30 fixed epochs. Finally, the trained network is tested for the year 2010.

Ignoring years older than four years prevent their noise effect but also reduces training set size. Experiments of this work show that embedding layer weights are learned better when the reports of training set are temporally closer to target year. Furthermore, embedding layer may be biased easier than convolution layer since convolution layer features are learned from larger structures (n-grams).

Training our model by using all years from 1996 to test year is time-consuming. Therefore, transfer learning is used to reduce the time consumption. Transfer learning is applied to the convolution layer by training its weights using the reports of the older years and using these pretrained weights as initial weights while training our model for the recent years. The relatedness of the transfer domains has a direct effect on the amount of improvement [66]. Since we use transfer learning for the same domain, its effect would be very high. First, convolution layer weights are trained by freezing the embedding layer which is initialized with pretrained word embeddings and using years 1996 to 2006 for 120 epoch with early stopping while other hyper-parameters are kept as described above. Later, convolution layer weights are initialized with weights which are pretrained by transfer learning. It results in reduction of time consumption.

5.2. Extended Models

Using transfer learning convolution layer, four different models are built. Since convolution layer weights are trained using pretrained word embeddings, those models perform well only when their embedding layers are initialized with pretrained word embeddings. Following [45], multichannel embedding layers are applied to some models.

- *CNN-STC:* A model with single channel non-static pretrained embedding layer and a transferred convolution layer which is static.
- *CNN-NTC:* Same as CNN-STC but its transferred convolution layer is non-static.
- *CNN-STC-multi:* A model with two channel of embedding layers, both are pretrained but one is static and other one is non-static. Transferred convolution layer is also static.
- *CNN-NTC-multi:* Same as CNN-STC-multichannel but its transferred convolution layer is non-static

We run our experiments on a GPU. Since training DL models consist of numerous matrix multiplication, running the algorithm on a GPU is much more faster than running it on a CPU. However, the algorithm has to be parallelized because GPUs are faster than CPUs if the algorithm is parallelized. Pytorch is used to parallelize our algorithm. Pytorch provides the abstraction to run the algorithm on GPU.

A single Nvidia Tesla V100 is used to train the DL models. Training and evaluating a CNN-NTC-multi model takes 1.5 to 3 hours. Since the number of epochs is not fixed, training duration differs. On the other hand, training the model for a single epoch does not vary. It takes 2 minutes to train CNN-NTC-multi on a Nvidia Tesla V100.

CNN-NTC-multi contains 2 channel of embedding layer and all layers are non-static. Therefore, training other models takes less time. For example, training CNN-STC model takes the least time because it has a single channel of embedding layer and it is static. Time cost of each epoch of a CNN-STC model is 30 seconds. A complete

training evaluation of CNN-STC costs 25 minutes to 40 minutes.

5.3. Results

Table 5.2 indicates that performance of our CNN-simple (baseline) model is comparable with EXP-SYN, the best model represented in [55], which uses a manually created lexicon and POS tagger. Furthermore, the best predictions for the years 2008 and 2010 are achieved by the CNN-simple model.

Our best model, CNN-NTC, decreases the average error by 10% and produces the best predictions for the last three years of the experiment. Note that, the convolution layer weights of each extended model is pretrained by using transfer learning. However, CNN-simple has randomly initialized convolution layer weights. Furthermore, each layer of CNN-simple is non-static.

Table 5.2. Performance of different models, measured by Mean Square Error (MSE).

Model	2008	2009	2010	2011	2012	2013	Avg
EXP-SYN [55]	0.6537	0.2387*	0.1514	0.1217	0.2290	0.1861	0.2634
CNN-simple	0.3716*	0.4708	0.1471*	0.1312	0.2412	0.2871	0.2748
CNN-STC	0.5358	0.3575	0.3001	0.1215	0.2164	0.1497	0.2801
CNN-NTC-multi	0.5077	0.4353	0.1892	0.1605	0.2116	0.1268	0.2718
CNN-STC-multi	0.4121	0.4040	0.2428	0.1574	0.2082	0.1676	0.2653
CNN-NTC	0.4672	0.3169	0.2156	0.1154*	0.1944*	0.1238*	0.2388*

Table 5.3 shows the coefficient of determination of each model for years between 2008 and 2013. Coefficient of determination is bounded between 0 and 1 and higher coefficient of determination indicates a better prediction of distribution. It can be easily seen that the coefficient of determination results differ from MSE results.

Better MSE performance indicates that the prediction is close to the target. However, it is not affected from the distribution of predictions. Therefore, a model which outputs a constant prediction value can perform better than a model which predicts most values correctly but miss others with a higher error. On the other hand, the coefficient of determination does not provide any information about the distance between prediction and real values but only the correctness of the distribution.

Table 5.3. Performance of different models, measured by coefficient of determination.

Model	2008	2009	2010	2011	2012	2013	Avg
CNN-simple	0.1755	0.0071	0.3548*	0.3296	0.4993*	0.4876	0.3089
CNN-STC	0.1720	0.2781*	0.3222	0.3274	0.4913	0.5157*	0.3511*
CNN-NTC-multi	0.1568	0.1981	0.2849	0.3239	0.4214	0.5065	0.3152
CNN-STC-multi	0.1364	0.1562	0.2308	0.2375	0.2530	0.3847	0.2331
CNN-NTC	0.1792*	0.2325	0.3283	0.3583*	0.4812	0.5057	0.3475

In Table 5.3 it can be seen that CNN-STC performs the best and CNN-STC-multi performs the worst. Furthermore, comparing the results in Table 5.2 and Table 5.3 shows that the performances of CNN-STC and CNN-STC-multi have the highest difference compared to others. Having a static convolution layer is the common property of CNN-STC and CNN-STC-multi.

The ranking performance of a model is valuable for some real world applications such as portfolio management. Furthermore, better ranking performance indicates that the label distribution is explained better. Table 5.4 shows the ranking performance of each model which is presented in this work, using Spearman's rank correlation coefficient.

Spearman's rank correlation coefficient is bounded between -1 and 1. Higher Spearman's rank correlation coefficient means the model captures larger proportion of variability in the labels. It can be seen that ranking performance of CNN-NTC

is as good as its regression performance. On the other hand, CNN-STC can model future distribution of stock return volatilities better than future values of stock return volatilities.

Table 5.4. Ranking performance of different models, measured by Spearman's rank correlation coefficient.

Model	2008	2009	2010	2011	2012	2013	Avg
CNN-simple	0.3884	0.0814	0.5758*	0.5842	0.7064	0.7060	0.5070
CNN-STC	0.3875	0.5226*	0.5570	0.5737	0.7149*	0.7341*	0.5816*
CNN-NTC-multi	0.3727	0.4293	0.5187	0.5625	0.6531	0.7332	0.5449
CNN-STC-multi	0.3424	0.4042	0.4641	0.4924	0.4945	0.6305	0.4713
CNN-NTC	0.3921*	0.4713	0.5500	0.5910*	0.6978	0.7234	0.5709

In all experiments, MSE is used as the loss function which means each model tries to optimize MSE. On the other hand, coefficient of determination and Spearman's rank correlation coefficient are reported only to evaluate the ranking performance of different models. Changing the loss function may improve ranking performance results and performance orders of the models.

5.4. Analysis

The embedding weights of CNN-NTC are compared with the pretrained word embeddings to determine the most changed words. While comparing the most changed word vectors, the words with yearly frequency less than 250 and more than 5000 are filtered out. Table 5.5 presents the top 10 most changed words and cosine distances to their pretrained vectors. Note that presented words are stemmed. Since words are in lowercase, the word *ETC* may cause confusion. It is an abbreviation and stands for Exchange-Traded Commodity which is a common word in finance domain and stemmed version includes its plural form *ETCs* also.

Table 5.5. Top-10 most changed words, extracted from non-static embedding layer.

Word	**Cosine Distance**
anoth	0.2565
concern	0.2436
etc	0.2431
accordingli	0.2353
entir	0.2349
stabil	0.2328
increment	0.2308
thu	0.2306
situat	0.2167
guaranti	0.2120

The stemmed words *concern*, *stabil* and *guaranti* are sentiment words and contained by finance sentiment lexicon [27]. Having 3 sentiment words out of 10 words shows that our model uses sentiment information but not solely depend on sentiment words.

We also analyzed the most changed sentiment word, *concern*, by extracting the 10 nearest words of pretrained word embeddings and CNN-NTC embedding weights separately (Table 5.6). It can be observed that *pertain*, *about* and *fear* are replaced with *safeti*, *trend* and *dmaa*.

Stem words *safeti* and *trend* are related with the stem word *concern*. The word *pertain* is semantically very close to the word *concern*, they are even used interchangeably sometimes. However, *concern* can be replaced with *pertain* only if it does not have any sentiment polarity. It can be seen that expanding the lexicon using word embeddings, like previous works did [14, 15, 55], can be problematic and may end up with a lexicon expansion containing semantically close but sentimentally far words.

Table 5.6. Top-10 most similar words to *concern* comparing their word vectors.

Static Embedding on 'concern'		**Non-static Embedding on 'concern'**	
Word	Cosine Distance	Word	Cosine Distance
regard	0.2772	regard	0.3233
privaci	0.5287	privaci	0.5433
inform	0.5587	safeti	0.5550
debat	0.5706	inform	0.5562
implic	0.5817	trend	0.5568
heighten	0.5825	heighten	0.5692
pertain	0.5844	inquiri	0.5959
about	0.5901	dmaa	0.6013
inquiri	0.5919	debat	0.6025
fear	0.5954	implic	0.6033

Another interesting word in the list is *DMAA*. It stands for dimethylamylamine which is an energy-boosting dietary supplement. In 2012, the U.S. Food and Drug Administration (FDA) warned DMAA manufacturers. In 10-K report of Vitamin Shoppe, Inc. published on February 26, 2013, concern of the company about DMAA is stated as shown in Figure 5.1.

It shows that the CNN model focuses on correct word features but can also overfit easier. In financial text regression task, the word *DMAA* is quite related with the word concern but it is not a common word and also sector specific.

As is common in the VMS industry, we rely on our third-party vendors to ensure that the products they manufacture and sell to us comply with all applicable regulatory and legislative requirements. In general, we seek representations and warranties, indemnification and/or insurance from our vendors. However, even with adequate insurance and indemnification, any claims of non-compliance could significantly damage our reputation and consumer confidence in our products. In addition, the failure of such products to comply with applicable regulatory and legislative requirements could prevent us from marketing the products or require us to recall or remove such products from the market, which in certain cases could materially and adversely affect our business, financial condition and results of operations. For example, products manufactured by third parties that contain 1.3-dimethylpentylamine/dimethylamylamine/13-dimethylamylamine (**DMAA**) are among our top selling products. Although we have received representations from our third-party vendors that these products comply with applicable regulatory and legislative requirements, media articles have suggested that **DMAA** may not comply with the Dietary Supplement Health and Education Act of 1994. On April 27, 2012, the U.S. Food and Drug Administration (FDA) announced that it had issued warning letters to ten manufacturers and distributors of dietary supplements containing **DMAA** for allegedly marketing products for which evidence of the safety of the product had not been submitted to FDA. The FDA also indicated that the warning letters advised the companies that the FDA is not aware of evidence or history of use to indicate that **DMAA** is safe. If it is determined that **DMAA** does not comply with applicable regulatory and legislative requirements, we could be required to recall or remove from the market all products containing **DMAA** and we could become subject to lawsuits related to any alleged non-compliance, any of which recalls, removals or lawsuits could materially and adversely affect our business, financial condition and results of operations. As a result of the indeterminable level of product substitution and reformulated product sales, we cannot reliably determine the potential impact of any such recall or removal on our business, financial condition or results of operation.

Figure 5.1. Part of 10-K report of Vitamin Shoppe, Inc. published on February 26, 2013.

6. CONCLUSION

In this thesis, the financial text regression task is described and various works which are related to the task are presented. Next, different datasets are reviewed in details. System architecture and model architecture of our work is presented. Later, detailed experimental setup is discussed. Extended models which has convolution layer weights trained previously are introduced. Results of three different performance measure are presented for six models and years 2008 to 2013. Resulting trained models are also analyzed.

We aimed to build a deep learning model which is not dependent to a financial sentiment lexicon while increasing the performance of the model. Importance of removing the financial sentiment lexicon dependency is due to the manual work cost of financial sentiment lexicon preparation. The work which is presented in this paper uses word embeddings to remove the financial sentiment lexicon dependency. Word embeddings are also used in previous studies. However, in the previous studies word embeddings are used to expand the lexicon without including word embeddings to the model. On the contrary, our work includes word embeddings directly to the model as main input.

In addition, transfer learning is applied to the convolution layer since effect of temporal information on distinct layers differs. Our model benefits from transfer learning since learning generic financial n-grams can take a long training phase. Nevertheless, training the convolution layer using a wide range of reports from different years provides generic financial n-grams embedded weights. MSE, coefficient of determination and Spearman's rank correlation coefficient results showed that freezing the transferred convolution layer weights can result in distinct results for different performance metrics. On the other hand, performance results of models with non-static convolution layer weights are more stable, independent from performance metric.

After the results are presented and discussed, static and non-static word vectors of multichannel models are analyzed. The analysis demonstrates that the CNN model tracks the sentiment polarity of the words successfully and it does not depend on sentiment words only. For a better demonstration, a stem and reports including the stem is presented. However, it is also observed that CNN models can overfit easier since the stem is a sector specific abbreviation.

This work can be improved by focusing on two different goals. First goal is performance improvement. Performance of the results can be improved by training word embeddings using more recent techniques and replacing the cost function with a fusion of MSE and Spearman's rank correlation coefficient. Second goal is expansion of the work. This work is focused on the text source and did not include any historical market data or any other metadata. Further research on including metadata to CNN model for the same task may increase the value and analysis.

REFERENCES

1. Lyons, J., *Natural Language and Universal Grammar: Essays in Linguistic Theory*, Vol. 1, Cambridge University Press, 1991.

2. Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, 2014.

3. Finkel, J. R. and C. D. Manning, "Nested Named Entity Recognition", *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 141–150, Aug. 2009.

4. Schmid, H., "Part-of-speech Tagging With Neural Networks", *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, 1994.

5. Kenyon-Dean, K., E. Ahmed, S. Fujimoto, J. Georges-Filteau, C. Glasz, B. Kaur, A. Lalande, S. Bhanderi, R. Belfer, N. Kanagasabai, R. Sarrazingendron, R. Verma and D. Ruths, "Sentiment Analysis: It's Complicated!", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1886–1895, Jun. 2018.

6. Fan, A., D. Grangier and M. Auli, "Controllable Abstractive Summarization", *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 45–54, Jul. 2018.

7. Wadhwa, S., K. Chandu and E. Nyberg, "Comparative Analysis of Neural QA models on SQuAD", *Proceedings of the Workshop on Machine Reading for Question Answering*, pp. 89–97, Jul. 2018.

8. Kogan, S., D. Levin, B. R. Routledge, J. S. Sagi and N. A. Smith, "Predicting

Risk from Financial Reports with Regression", *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 272–280, Jun. 2009.

9. Lampos, V. and N. Cristianini, "Tracking the flu pandemic by monitoring the social web", *2010 2nd International Workshop on Cognitive Information Processing*, pp. 411–416, June 2010.

10. Joshi, M., D. Das, K. Gimpel and N. A. Smith, "Movie Reviews and Revenues: An Experiment in Text Regression", *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 293–296, Jun. 2010.

11. Nguyen, D., N. A. Smith and C. P. Rosé, "Author Age Prediction from Text using Linear Regression", *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pp. 115–123, Jun. 2011.

12. Lampos, V., D. Preoţiuc-Pietro and T. Cohn, "A user-centric model of voting intention from Social Media", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 993–1003, Aug. 2013.

13. Wang, C.-J., M.-F. Tsai, T. Liu and C.-T. Chang, "Financial Sentiment Analysis for Risk Prediction", *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 802–808, Oct. 2013.

14. Tsai, M.-F. and C.-J. Wang, "Financial Keyword Expansion via Continuous Word Vector Representations", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1453–1458, Oct. 2014.

15. Rekabsaz, N., M. Lupu, A. Baklanov, A. Dür, L. Andersson and A. Hanbury, "Volatility Prediction using Financial Disclosures Sentiments with Word

Embedding-based IR Models", *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1712–1721, Jul. 2017.

16. Tetlock, P. C., M. Saar-Tsechansky and S. Macskassy, "More Than Words: Quantifying Language to Measure Firms' Fundamentals", *The Journal of Finance*, Vol. 63, No. 3, pp. 1437–1467, 2008.

17. Nuij, W., V. Milea, F. Hogenboom, F. Frasincar and U. Kaymak, "An Automated Framework for Incorporating News into Stock Trading Strategies", *IEEE Trans. on Knowl. and Data Eng.*, Vol. 26, No. 4, pp. 823–835, Apr. 2014.

18. Kazemian, S., S. Zhao and G. Penn, "Evaluating Sentiment Analysis Evaluation: A Case Study in Securities Trading", *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 119–127, Jun. 2014.

19. Ding, X., Y. Zhang, T. Liu and J. Duan, "Deep Learning for Event-driven Stock Prediction", *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 2327–2333, 2015.

20. Narayanan, R., B. Liu and A. Choudhary, "Sentiment Analysis of Conditional Sentences", *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 180–189, Aug. 2009.

21. Nguyen, T. H. and K. Shirai, "Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1354–1364, Jul. 2015.

22. Bar-Haim, R., E. Dinur, R. Feldman, M. Fresko and G. Goldstein, "Identifying and Following Expert Investors in Stock Microblogs", *Proceedings of the 2011 Confer-*

ence on Empirical Methods in Natural Language Processing, pp. 1310–1319, Jul. 2011.

23. Nopp, C. and A. Hanbury, "Detecting Risks in the Banking System by Sentiment Analysis", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 591–600, Sep. 2015.

24. Drucker, H., C. J. C. Burges, L. Kaufman, A. J. Smola and V. Vapnik, "Support Vector Regression Machines", *Advances in Neural Information Processing Systems 9*, pp. 155–161, 1997.

25. Joachims, T., *Advances in Kernel Methods*, MIT Press, Cambridge, MA, USA, 1999.

26. Scholkopf, B. and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA, 2001.

27. Loughran, T. and B. McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks", *The Journal of Finance*, Vol. 66, No. 1, pp. 35–65, 2011.

28. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", *Advances in Neural Information Processing Systems 26*, pp. 3111–3119, 2013.

29. Wolpert, D. H., "Stacked Generalization", *Neural Networks*, Vol. 5, pp. 241–259, 1992.

30. Schölkopf, B., K. Tsuda and J. Vert, "Support Vector Machine Applications in Computational Biology", *Kernel Methods in Computational Biology*, 2004.

31. Hacısalihzade, S., *Control Engineering and Finance*, Lecture Notes in Control and

Information Sciences, Springer International Publishing, 2017.

32. Doucet, A. and H. Ahonen-Myka, "Non-Contiguous Word Sequences for Information Retrieval", *Second ACL Workshop on Multiword Expressions: Integrating Processing*, pp. 88–95, Jul. 2004.

33. Bengio, Y., R. Ducharme, P. Vincent and C. Janvin, "A Neural Probabilistic Language Model", *J. Mach. Learn. Res.*, Vol. 3, pp. 1137–1155, Mar. 2003.

34. Levy, O. and Y. Goldberg, "Linguistic Regularities in Sparse and Explicit Word Representations", *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pp. 171–180, Jun. 2014.

35. Goldberg, Y. and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method", *CoRR*, Vol. abs/1402.3722, 2014.

36. Mikolov, T., G. Corrado, K. Chen and J. Dean, "Efficient Estimation of Word Representations in Vector Space", pp. 1–12, 01 2013.

37. Pennington, J., R. Socher and C. Manning, "Glove: Global Vectors for Word Representation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Oct. 2014.

38. Bojanowski, P., E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

39. Décary, M. and G. Lapalme, "An Editor for the Explanatory and Combinatory Dictionary of Comtemporary French (DECFC)", *Computational Linguistics*, Vol. 16, No. 3, pp. 145–154, 1990.

40. Mel'čuk, I. and A. Polguere, "A Formal Lexicon in Meaning-Text Theory (Or How to Do Lexica with Words)", *Computational Linguistics*, Vol. 13, pp. 261–275, 1987.

41. J. Stone, P., D. C. Dunphy, M. Smith and D. M. Ogilvie, "The General Inquirer: A Computer Approach to Content Analysis", *American Educational Research Journal - AMER EDUC RES J*, Vol. 4, 01 1966.

42. Yih, W.-t., X. He and C. Meek, "Semantic Parsing for Single-Relation Question Answering", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 643–648, Jun. 2014.

43. Shen, Y., X. He, J. Gao, L. Deng and G. Mesnil, "Learning Semantic Representations Using Convolutional Neural Networks for Web Search", *Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374, 2014.

44. Kalchbrenner, N., E. Grefenstette and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 655–665, Jun. 2014.

45. Kim, Y., "Convolutional Neural Networks for Sentence Classification", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Oct. 2014.

46. Bitvai, Z. and T. Cohn, "Non-Linear Text Regression with a Deep Convolutional Neural Network", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 180–185, Jul. 2015.

47. Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural Language Processing (Almost) from Scratch", *J. Mach. Learn. Res.*, Vol. 12, pp. 2493–2537, Nov. 2011.

48. Le, H. T., C. Cerisara and A. Denis, "Do Convolutional Networks need to be Deep for Text Classification ?", *AAAI Workshops*, 2018.

49. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

50. Händschke, S. G., S. Buechel, J. Goldenstein, P. Poschmann, T. Duan, P. Walgenbach and U. Hahn, "A Corpus of Corporate Annual and Social Responsibility Reports: 280 Million Tokens of Balanced Organizational Writing", *Proceedings of the First Workshop on Economics and Natural Language Processing*, pp. 20–31, Jul. 2018.

51. U.S.-SEC, *Filings and Forms*, 2017, `https://www.sec.gov/edgar.shtml`, accessed at April 2019.

52. Griffin, P. A., "Got Information? Investor Response to Form 10-K and Form 10-Q EDGAR Filings", *Review of Accounting Studies*, Vol. 8, No. 4, pp. 433–460, Dec 2003.

53. Kutcher, L., E. Peng and K. Zvinakis, "The Impact of the Accelerated Filing Deadline on Timeliness of 10-K Filings", *Journal of Accounting and Public Policy*, 07 2007.

54. Campbell, J. L., H. Chen, D. S. Dhaliwal, H.-m. Lu and L. B. Steele, "The information content of mandatory risk factor disclosures in corporate filings", *Review of Accounting Studies*, Vol. 19, No. 1, pp. 396–455, Mar 2014.

55. Tsai, M.-F., C.-J. Wang and P.-C. Chien, "Discovering Finance Keywords via Continuous-Space Language Models", *ACM Trans. Manage. Inf. Syst.*, Vol. 7, No. 3, pp. 7:1–7:17, Aug. 2016.

56. Fama, E. F. and K. R. French, "Common risk factors in the returns on stocks and bonds", *Journal of Financial Economics*, Vol. 33, No. 1, pp. 3 – 56, 1993.

57. Fahlman, E. and E. Pettersson, "Media Coverage and Abnormal Trading Volume", p. 39, 2017.

58. Yahoo, *Yahoo Finance*, 2019, `https://finance.yahoo.com`, accessed at April 2019.

59. Routledge, B., S. Kogan, J. Sagi and N. Smith, *10-K Corpus*, 2009, `http://www.cs.cmu.edu/ ark/10K/`, accessed at April 2019.

60. CFDA and CLIP, *FIN10K*, 2017, `https://clip.csie.org/10K/data`, accessed at April 2019.

61. ADmIRE, *Index of / admire/financialvolatility/*, 2017, `http://ifs.tuwien.ac.at/ admire/financialvolatility`, accessed at April 2019.

62. Bao, Y. and A. Datta, "Summarization of Corporate Risk Factor Disclosure through Topic Modeling", *ICIS*, 2012.

63. Walgenbach, P. and U. Hahn, *Jena Organization Corpus (JOCo)*, 2018, `https://www.orga.uni-jena.de/en/Jena+Organization+Corpus+(JOCo)`, accessed at April 2019.

64. Schneider, J., *The Extrasolar Planets Encyclopaedia*, 2010, `http://python.org`, accessed at April 2011.

65. Porter, M., "An Algorithm for Suffix Stripping", *Program: Electronic Library and Information Systems*, Vol. 14, 03 1980.

66. Yang, Z., R. Salakhutdinov and W. W. Cohen, "Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks", *CoRR*, Vol. abs/1703.06345, 2017.